

Implantando uma arquitetura de serviços RESTful no CPD/UFSM

Universidade Federal de Santa Maria

Centro de Processamento de Dados

Everton de Vargas Agilar (UnB / UFSM)

Jader Adiel (UFSM)

Agenda

- Barramento de serviços ERLANGMS
 - Objetivo do projeto
 - Histórico do projeto
 - Design e princípios da arquitetura
 - Artigos produzidos para divulgar o barramento
- Implantação do barramento na UFSM
 - ► Panorama geral sobre modernização
 - O que já foi feito
 - O que ainda precisa ser feito x prioridade
 - Principais desafios do trabalho
 - Caso prático 1: Web services implementados para o NCC
 - Caso prático 2: TClis do SIE invocando web services Java



Barramento de serviços ERLANGMS



Objetivo do projeto

ERLANGMS

É um projeto desenvolvido inicialmente na UnB para auxiliar a modernização de sistemas para uma arquitetura orientada a serviços por meio de um barramento de serviços + SDK + processo de modernização e arquitetura documentado (SMSOC).



UFSM

Universidade Federal de Santa Maria

Histórico do projeto

2015

2016 2017 2018

2019

Primeira versão básica de um barramento REST em Erlang.

simples para

desenvoyler

usando os

Foi escrito um SDK

servicos em Java

O barramento recebeu suporte para HTTP 2.0 e HTTPS

realizar filtros mais

A capacidade de

melhor os tipos de

serialização foi

complexos.

dados.

de atributo Foco em A interface da API foi melhorada com a possibilidade de

desempenho com a Implementação de cache de queries usando named queries da JPA.

servidor de

aplicação.

A API passou a

suportar operadores

Implementação do ResultCaché para cachear o resultado do webservice no os operadores. barramento e evitar que requisições iquais seiam enviadas para o

O SDK recebeu diversas melhorias de performance e estabilidade

Foco na segurança com suporte para dois protocolos: OAuth2 e LDAP v3.

Foi incluído a possibilidade de gerar consultas com SQL nativo usando

A SDK está sendo redesenhada para ser mais agnóstica e conseguir trabalhar com Sprint e Java EE.

O operador filter está sendo ampliado para suportar condições com and/or e precedência de condições.

Está sendo implementado suporte para API constraints para limitar o uso da API em cada endpoint.

A serialização de dados está sendo refinada para fazer a conversão de entidades e value objects (VO).

operadores filter, fields, sort e limit. Estudo de caso para demonstrar o uso da REST API.

ampliada para tratar

5/17



Histórico do projeto

Principais features em uso na UnB

Features

- ► HTTP/2, HTTP/1.1, HTTPS (Secure TLS Listener);
- ► LDAP v3 Proxy LDAP;
- ► HTTP Basic authentication;
- OAuth2 authentication;
- ► SDK para criar serviços em Java;
- ► Query Api filter, fields, sort, limit.
- ► Suporte para Dados Abertos.



Design e princípios da arquitetura

Design e princípios da arquitetura

- Multiplataforma e arquitetura modular;
- Serviços especificados em catálogos de serviços;
- Cluster de serviços para evitar ponto único de falhas;
- ► Modelo de concorrência para serviços em Erlang: Actor Model;
- ► Baixa curva de aprendizado.



Artigos produzidos para divulgação do projeto

- ► A systematic mapping study on legacy system modernization 2016
- Uma abordagem orientada a serviços para a modernização de sistemas legados – 2016
- Uma Implementação do Protocolo OAuth 2 em Erlang para uma Arquitetura Orientada a Serviço – 2017
- ► An experience report on the adoption of microservices in three brazilian government institutions 2018
- ▶ Melhoria na Publicação de Dados Abertos: Automatização na Publicação e Indexação Semântica dos Dados - 2018
- ► Integração das Bases de Login e Senha dos Sistemas da Universidade de Brasília - UnB - 2019 8/17



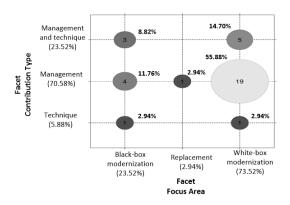
Implantação do barramento na UFSM



UFSMUniversidade Federal de Santa Maria

Panorama geral sobre modernização

Estudo realizado na UnB para identificar as estratégias de modernização





O que já foi feito

O que já foi feito

- ► Implantação do barramento em partes, priorizando:
 - Query API
 - Subsistema de serialização;
 - Adaptar a interfaces do SDK para trabalhar de forma agnóstica com Java EE e Spring.
- Features visando a migração de backends SIE para Java:
 - ► Novo operador format para serializar dados como Dataset;
 - Novo subsistema provider no SDK para permitir seu uso independente da camada de negócio;
 - Provider RestApiMetaQueryProvider;
 - ► Cliente REST implementado no Delphi com suporte a SSL.



O que ainda precisa ser feito x prioridade

O que ainda precisa ser feito x prioridade

- Tornar a camada de fachada mais simples Moderado;
- Implementar o subsistema de autenticação OAuth2 (ou integrar por completo o barramento) – Alto;
- ► Analisar os principais métodos do TCliBusiness e prover um suporte adequado no SDK para facilitar a migração Alto;
- Permitir gerar uma consulta por meio da Query API para buscar dados de um atributo lista de um objeto principal – Baixo;
- ▶ Permitir encadear funções nos atributos das entidades com o operador % para modificar a forma como o dado é lido da fonte de dados – Baixo.



Principais desafios do trabalho

Principais desafios do trabalho

- Levando em consideração o custo de tempo para extrair as regras de negócios comuns automatizadas no core do SIE, estamos preferindo focar mais nas TClis dos frontends;
- ► A quantidade de herança e, principalmente, métodos virtuais nas classes bases do pkCDESP tornam a migração mais demorada;
- ► Há exceções, as meta-queries do Delphi podem ser convertidas facilmente em webservices como o SDK do barramento;



Principais desafios do trabalho

Principais desafios do trabalho

- Para o cliente REST do frontend Delphi obter os dados no formato Dataset para os TClis, é necessário especificar o schema dos dados. É relativamente fácil mas trabalhoso;
- Pelo menos por enquanto, pelo fato do SIE CORE ser um artefato monolítico em termos de unidade de deploy, será bem mais desafiador o uso de microservices e Docker no futuro.



Caso prático 1: Web services implementados para o NCC





Caso prático 2: TClis do SIE invocando web services Java





UFSMUniversidade Federal de Santa Maria

Obrigado!